REMARKS

In an office action dated May 7, 2004, the Examiner rejected claims 1-2, 4-6, 8-9, 11-13, 15-16 and 18-20 under 35 U.S.C. §103(a) as obvious over Sher et al. (US Patent 5,668,751) in view of Gee et al. (US Patent 6,374,286). Claims 3, 7, 10, 14, 17 and 21 were objected to as dependent on rejected base claims, but otherwise indicated to contained allowable subject matter.

Claims 3, 7, 10, 14, 17 and 21 have been re-written in independent form, incorporating all the limitations of the respective claims from which they originally depended. As amended, these claims are therefore allowable.

Independent claims 1, 8 and 15 have been amended to clarify certain aspects of the present invention. In particular, claims 1, 8 and 15 have been amended to clarify that the claimed method and apparatus is a data processing method and apparatus, in which arrays (data) are algorithmically processed by a digital data processing device, and to further clarify that one of potentially multiple "machine codes" (i.e., executable code sequences) is selected for executing an algorithmic process according to the state of the flags. As amended, the claims are patentable over the cited art.

Applicants' invention relates to the optimization of computer processes performed on certain arrays. Generally, it is possible to realize substantial execution performance efficiencies if it can be known in advance that the array will conform to a some regular structure, e.g., the array occupies a contiguous area of memory. However, it is often difficult to know this at compile time, particularly in an object-oriented language. Several prior art approaches are discussed in applicant's background, but none is completely satisfactory. In accordance with applicants' invention, flags are maintained during execution to indicate whether a multidimensional array has the requisite regular structure necessary for performing a code optimization. If the flag indicates

that the array conforms to the required regular structure, the optimized code is executed; if not, a more general case, but less optimized, code is executed. The optimized code could be generated in advance by a static compiler, or could be generated by a "Just In Time" (JIT) compiler.

Sher, cited by the Examiner, discloses a method for programming an array of antifuses in a circuit. According to *Sher*, a respective flag is maintained for each antifuse to indicate whether it has been sufficiently set, and if so, current to that antifuse is shut off. The purpose of *Sher*'s invention is to avoid damaging the circuit with excessive current, and to obtain a circuit with a more consistent range of resistances. The secondary reference, *Gee*, is cited to show the use of multidimensional arrays, but otherwise does not teach or suggest any particular array operations.

The Examiner read the original claims broadly to encompass a process performed on an array of hardware devices. In order to clarify the nature of their invention, applicants' have amended independent claims 1, 8 and 15. Representative amended claim 1 recites:

> 1. A method for processing a multidimensional array object comprising array objects, said *multidimensional array object and said array objects being digital data objects storable in addressable data storage locations of a digital data processing device*, said method comprising the steps of:
>
> managing flags for said multidimensional array object, said flags representing whether it is possible to optimize a process for elements of said multidimensional array object, said process being a defined set of instructions executable by said digital data processing device; and
>
> executing a machine code performing said process, said *machine code being selected from among a plurality of machine codes performing said process according to a state of said flags*. [emphasis added]

Amended claims 8 and 15, while not identical in scope, contain limitations analogous to those italicized above.

As amended, the claims recite that *the array is data*, i.e., it is something stored in the addressable data storage locations (e.g., memory, disk storage, etc.) of a digital data processing

device. *Sher*'s array is an array of hardware devices. But more importantly, the machine code for executing an algorithmic process is selected from among multiple machine codes according to the state of the flags.[1] *Sher* discloses that the flags are used to shut off current (the current being controlled by a digital data processing device), but there is no disclosure of using different sets of executable code depending on the state of the flag. I.e., *Sher* has just one executable program, which uses the flags in a conventional manner to either drive or shut off current to the antifuses.

The essential point of applicants' invention is that *different executable code sequences* can be selected to perform *the same process* on a multidimensional array, and that a code sequence is selected according to the state of the flags. It is true that *Sher* does something based on the state of the flags, but what it does is to perform one of two different functions (drive current or shut it off). In this respect, *Sher* is similar to countless conventional programs, which may take a branch in the code depending on the state of a flag. In Sher, as in these other conventional programs, something functionally different is performed, depending on the state of the flag. In applicant's invention, *the same process* is performed in any case, and the end result of the process will be independent of the state of the flags, although different machine code will be executed. There is nothing in either *Sher* or *Gee*, alone or in combination, that would suggest this feature of applicants' invention.

In view of the foregoing, applicants submit that the claims are now in condition for allowance and respectfully request reconsideration and allowance of all claims. In addition, the

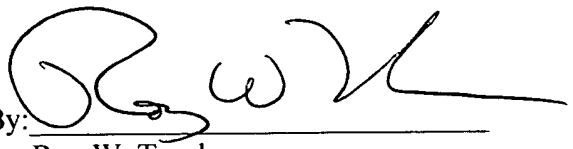
---

[1] The claims do not necessarily require that multiple machine code versions be compiled and in existence at the time the selection is made. If the code is statically compiled (i.e., before execution), this would generally be the case. However, if a JIT compiler is used, the run time module may examine the state of the flags, and generate the appropriate machine code depending on the state of the flags.

Examiner is encouraged to contact applicants' attorney by telephone if there are outstanding issues left to be resolved to place this case in condition for allowance.

Respectfully submitted,

TATSUSHI INAGAKI, et al.

By: 

Roy W. Truelson
Registration No. 34,265

Telephone: (507) 289-6256